CLOUD FUNDAMENTALS, ADMINISTRATION AND SOLUTION ARCHITECT

VIRTUALIZAÇÃO

FILIPI PIRES, CHRISTIANE DE PAULA REIS E HENRIQUE POYATOS

01

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 VIRTUALIZAÇÃO

O sistema operacional é como um grande “orquestrador” de cada um destes componentes, decidindo como otimizá-los para atender da melhor maneira possível as demandas oriundas dos processos ativos (ou programas em execução).

No entanto, se o sistema operacional está limitado aos recursos físicos do *hardware*, como explicar os modernos ambientes na nuvem, da qual arrastamos uma barra horizontal, e em um passe de mágica (e por uma centena de dólares), temos mais processador, memória e espaço de armazenamento?

Neste capítulo, abrimos a cortinas e relevamos como a mágica é feita, a partir de uma técnica conhecida como Virtualização.

1.1 Como funciona?

A virtualização surgiu como uma forma de separar melhor o *hardware* do *software*. Conforme mencionado, no modelo tradicional os *softwares* são instalados em um sistema operacional que, por sua vez, está instalado sobre uma infraestrutura física (o *hardware*). No entanto, sempre que os *softwares* demandassem por mais infraestrutura física para suportar mais usuários ou serviços, o *hardware* precisava ser substituído por outro mais poderoso (processadores mais rápidos, mais memória e por aí vai), o que é chamado de *upgrade* vertical.

Os *clusters* de computadores (que são computadores ligados entre si dividindo as tarefas e “juntando forças” para resolver problemas computacionais mais complexos) inauguraram a era *upgrades* horizontais: assim, sempre que fosse necessário atender mais usuários ou trabalhar com um volume maior de dados, bastava adicionar novos computadores a esta verdadeira “força tarefa”. Ainda neste modelo, cada um dos computadores colocados neste *cluster* mantinha sua própria estrutura de sistema operacional e, portanto, seus próprios processos.
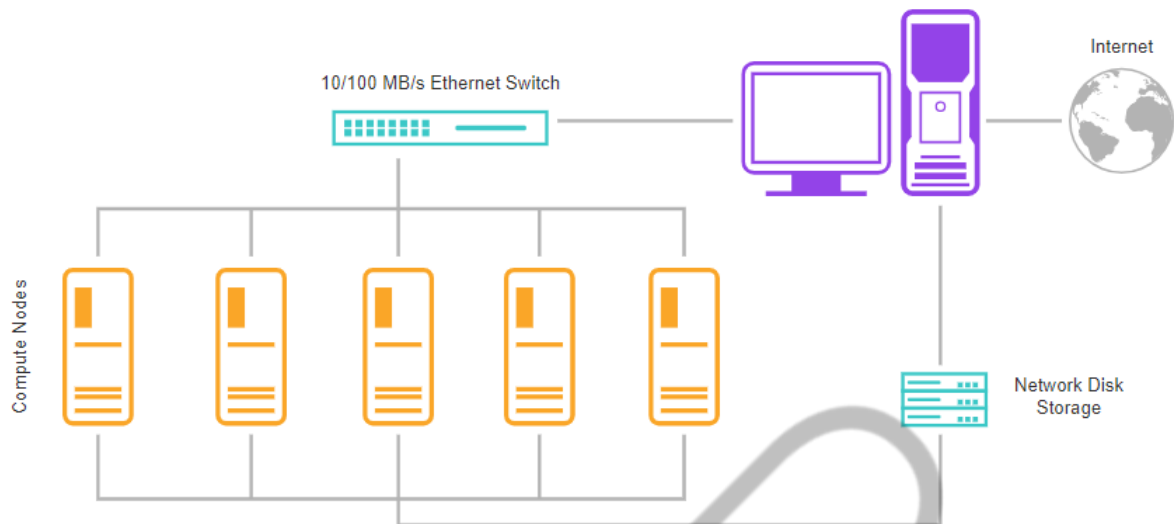
Figura 1.1 – *Cluster* de computadores
Fonte: Google Imagens (2019)

A virtualização possibilitou a abstração do *hardware*, pois uma camada de *software* **mimetizando uma infraestrutura física** é inserida entre o *hardware* (real) o sistema operacional, possibilitando uma separação mais eficiente destas duas camadas. A virtualização permite, por exemplo, a instalação de vários sistemas operacionais em um mesmo equipamento.
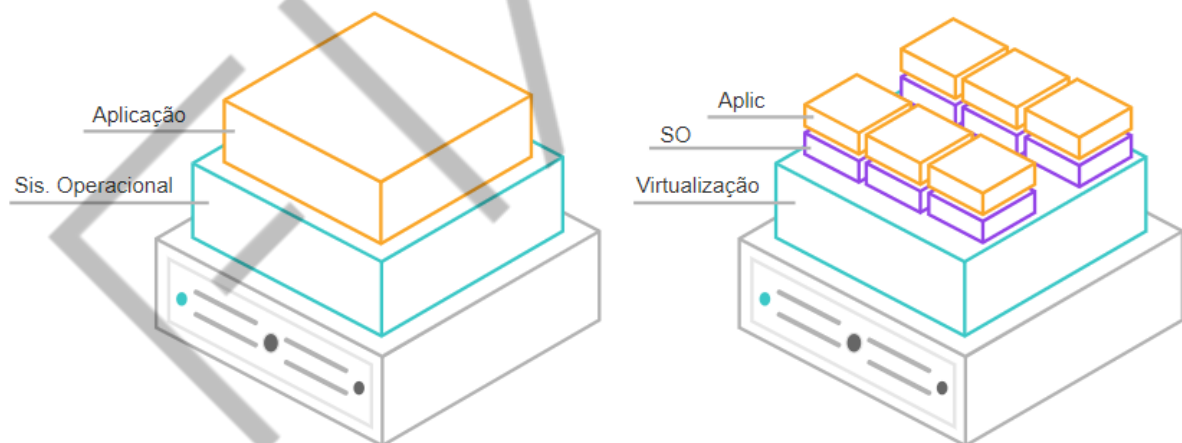
Figura 1.2 – Estrutura tradicional x Virtualização
Fonte: Google Imagens (2019)

Não demorou muito para surgir um inverso: Um único sistema operacional (chamado neste contexto de **máquina virtual**) instalada sob uma única camada de virtualização que compreende diversos computadores abaixo dela (o *cluster*!). Nossa máquina virtual "enxerga" um único *hardware* (que se comporta como um único supercomputador com vários núcleos e muita, muita memória para armazenamento), permitindo uma **flexibilidade muito grande no aumento destes recursos para a máquina virtual**. Sempre que esta precisar atender mais usuários, poderá receber

rapidamente mais processamento e memória e, por sua vez, sempre que a infraestrutura física for insuficiente, novos computadores serão adicionados ao *cluster* em um *upgrade* horizontal.

O *Cloud Computing* é o compartilhamento desta infraestrutura entre diversas necessidades e clientes, em um modelo de negócio e cobrança diferenciado. Mas isso já é um papo para outro capítulo.
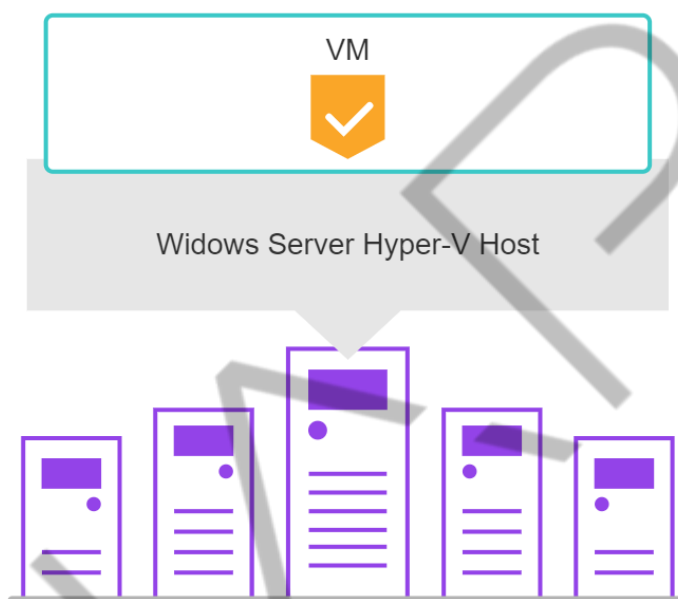
Figura 1.3 – Estrutura tradicional x Virtualização (2)
Fonte: Google Imagens adaptado por FIAP (2019)

1.2 Benefícios

Esta tecnologia permite simular aplicações, servidores, armazenamento e redes, sendo capaz de reduzir custos, aumentar eficiência, agilidade, flexibilidade e escalabilidade dentro de uma infraestrutura corporativa.

Com a virtualização é possível consolidar servidores, cargas de trabalho e ambientes, aumentar a utilização de recursos e acelerar a implantação de *desktop* e aplicativos.

As principais vantagens são:

- **Redução de custos:**

A virtualização pode reduzir investimentos em hardware, energia e espaço físico, isso porque permite reaproveitar recursos físicos que ficam ociosos.

Lembre-se que o processador, memória e outros recursos não são 100% exigidos o tempo todo, até porque alguns processos exigem mais processamento (chamados de *CPU bound*) e outros exigem mais das memórias (chamados de *I/O bound*), como os bancos de dados. Se em um exemplo hipotético a maioria dos processos de um sistema operacional exige, no máximo, 40% do processador, temos, então, 60% do processador "sobrando", certo?

A virtualização permite que um segundo sistema operacional ocupe os 60% restantes do processador sem qualquer prejuízo para os processos do primeiro. Sensacional, não é mesmo?
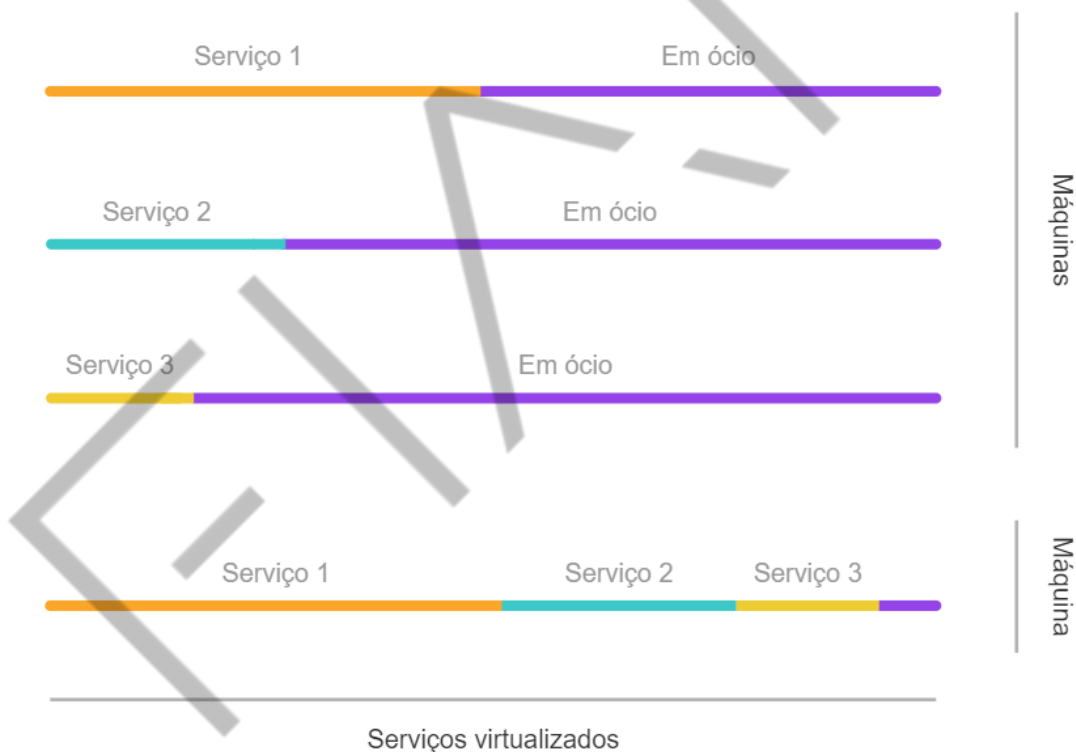
Figura 1.4 – Benefícios da Virtualização
Fonte: Infowester (s.d.)

- **Redução do tamanho do parque de equipamentos (*datacenter*):**

Com o melhor aproveitamento dos recursos atuais, a necessidade de aquisição de novos equipamentos diminui, reduzindo gastos com instalação, refrigeração, espaço físico, manutenção, consumo de energia e assim por diante.

- **Gerenciamento centralizado:**

Torna-se mais fácil monitorar serviços/processos em execução, já que o gerenciamento destes é feito de forma centralizada (no entanto, isso depende da estratégia de virtualização adotada).

- **Manutenção de sistemas legados:**

Um grande desafio nas grandes empresas é, sem dúvida alguma, a manutenção de sistemas legados. Não é raro encontrar um equipamento cujo *hardware* possui mais de vinte anos de idade, rodando um sistema operacional absolutamente obsoleto com o único objeto de manter “no ar” uma aplicação ou serviço que “só roda naquela configuração específica”.

Pois bem, a virtualização permite simular aquele *hardware* que já deveria ser sido aposentado há muito tempo, e instalar nesta infraestrutura emulada aquele Windows 3.11 (pergunte a seus pais) para fazer uma hora extra que até a Microsoft ficaria espantada. E tudo isso em componentes físicos que vão mergulhar no sono eterno a qualquer momento.

- **Ambientes de testes:**

A virtualização permite até mesmo emular hardwares de arquiteturas diferentes. Como testar uma aplicação feita para funcionar em *smartphone* Android ou um iPhone? Claro que você pode comprar os equipamentos, mas nem sempre é uma opção: um desenvolvedor de aplicativos preocupado com seus usuários teria que possuir vinte *smartphones* diferentes (chutando baixo). Por meio da virtualização, é possível simular uma arquitetura ARM (*Advanced RISC Machine*), típica de *smartphones*, em uma arquitetura x86-64 bits (dos microcomputadores).

- **Confiabilidade e Segurança:**

Já que cada máquina virtual (VM) funciona de maneira independente, se um problema surgir em uma delas (como uma vulnerabilidade de segurança, por exemplo) este não afetará as demais;

- **Migrações e ampliações mais simples:**

Trocar um fornecedor da infraestrutura *Cloud Computing* é mais simples, pois a VM pode ser exportada e importada em outro ambiente com facilidade (a alternativa

seria montar o sistema todo de novo, “do zero”). Além disso, ampliar o hardware demandando mais processamento, memória ou armazenamento se torna mais flexível, graças ao compartilhamento de *hardware* e *upgrades* horizontais.

1.3 Infraestrutura

Pode-se dizer que uma Máquina Virtual (ou *Virtual Machine*, VM), nada mais é que um computador emulado, que possui processador, memória, armazenamento, rede, interfaces (como a USB) e até periféricos emulados, com a capacidade de executar as mesmas funções que um computador físico.
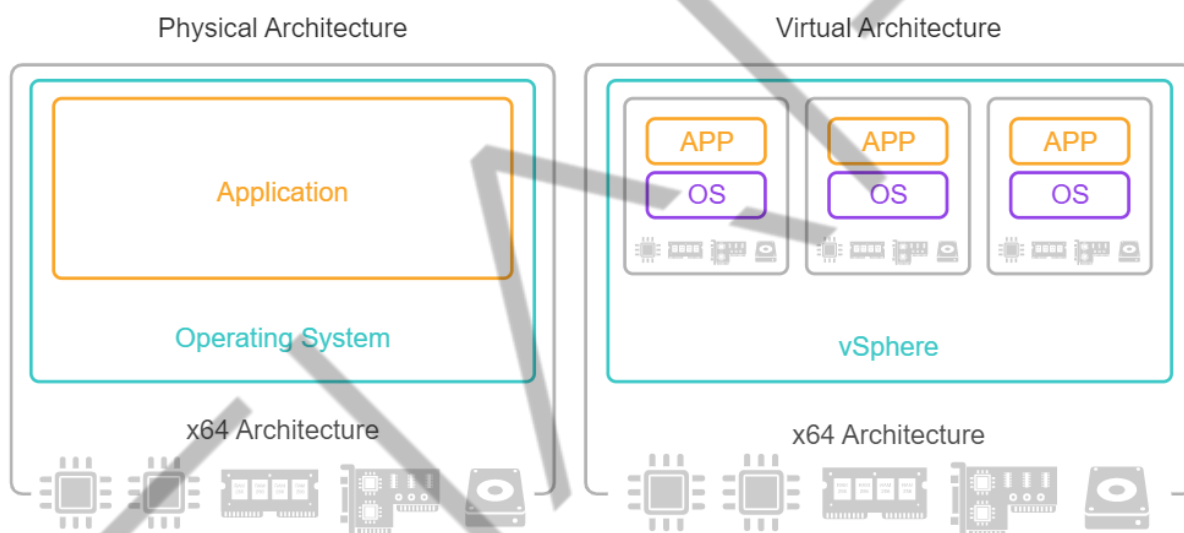
Figura 1.5 – Hardware
Fonte: Google Imagens (2019)

A virtualização é uma tecnologia de software já consagrada que muda fundamentalmente a maneira da computação em tirar componentes de camadas e consolidar recursos em *pools.* Podendo executar várias máquinas virtuais em uma única máquina física assim como vários Sistemas Operacionais (Windows, Linux, Solaris, entre outros) além de compartilhar os recursos desse computador único (CPU, memória, dispositivo de rede, armazenamento, este software que permite criar e executar máquinas virtuais é chamado de ***Hypervisor***.

Com a Virtualização podemos fornecer uma versão virtual de muitas tecnologias essenciais em computação, como principais podemos citar **Hardware**, **Armazenamento** e **Redes**:

- **Hardware**: esse é o um dos principais itens dentro da tecnologia de virtualização, um sistema operacional pode ser instalado sobre outro tipo de sistema, com seus recursos de *hardware* sendo representados via *software*.
- **Armazenamento**: também conhecido pelo nome de ***Software Defined Storage – SDS*** **(Armazenamento Definido por Software).** É uma camada de software criada sobre discos físicos, na qual os dispositivos acessam esses discos de modo a tornar o acesso mais flexível, gerenciável e personalizável.
- **Rede**: Chamado de ***Software Defined Networking – SDN*** **(Rede Definida por Software).** É possível criar um tipo de infraestrutura lógica (software) de redes sobre uma determinada rede física, permite a configuração e o detalhamento de acordo com as necessidades do ambiente.

Ao usar essas técnicas de virtualização, todos os dispositivos físicos podem ser representados em forma de softwares: servidores e estações de trabalho se tornam Máquinas Virtuais (VMs/*Virtual Machines*), a rede e o *storage* são virtualizados, transformando-se em **SDN e SDS** respectivamente. Com isso, construímos o que conhecemos por e **SDDC –** ***Software Defined Data Center*** **(*Data Center* Definido por *Software*).**
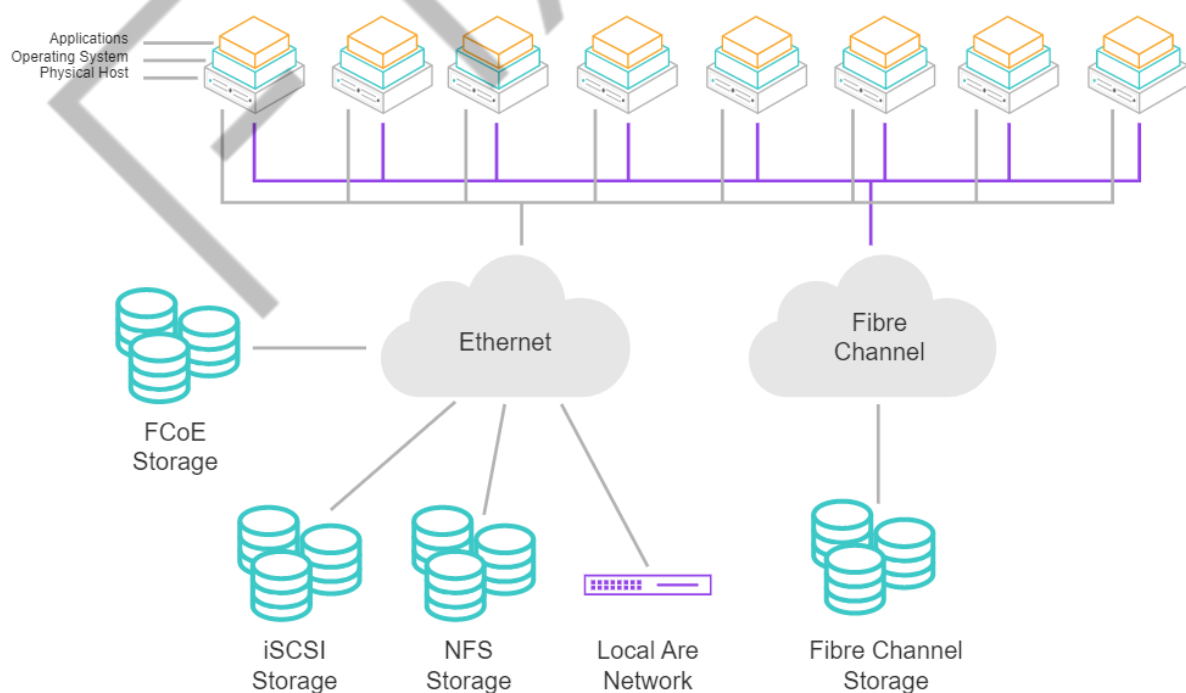
Figura 1.6 – Virtualização
Fonte: Google Imagens (2019)

Vamos convencionar alguns termos a partir de agora:

- **Hospedeiro (*host*):** trata-se da infraestrutura real, local da qual as VMs ficaram hospedadas; ele pode ter seu próprio sistema operacional (instalado diretamente no hardware, de forma convencional) ou não.
- ***Hypervisor*:** é uma camada de *software* localizada entre a camada de *hardware* e o sistema operacional.
- **Convidado ou hóspede (*guest*):** é a máquina (VM) instalada sob o *hardware* emulado.

1.4 Tipos de Virtualização

Existem vários tipos de virtualização. Vamos classificá-las a seguir.

1.4.1 Virtualização de aplicativos

A virtualização de aplicativos fornece um aplicativo hospedado em uma única máquina para um grande número de usuários. O aplicativo pode estar situado na nuvem em máquinas virtuais de alta qualidade, mas, como um grande número de usuários o acessa, seus custos são compartilhados por esses usuários.

Isso torna o aplicativo mais barato para entregar ao usuário final. O usuário final não precisa ter *hardware* de alta qualidade para executar o aplicativo; uma máquina barata, como uma estação de trabalho de baixo custo ou um terminal *thin client*, será suficiente.

E se os dados usados pelo aplicativo virtual são armazenados na nuvem, o usuário não está ligado a nenhum dispositivo ou local para usar a aplicação ou acessar seus dados. Normalmente, nesses casos, o aplicativo virtual é consumido por meio de um aplicativo móvel ou de um navegador da Internet pelo usuário final.
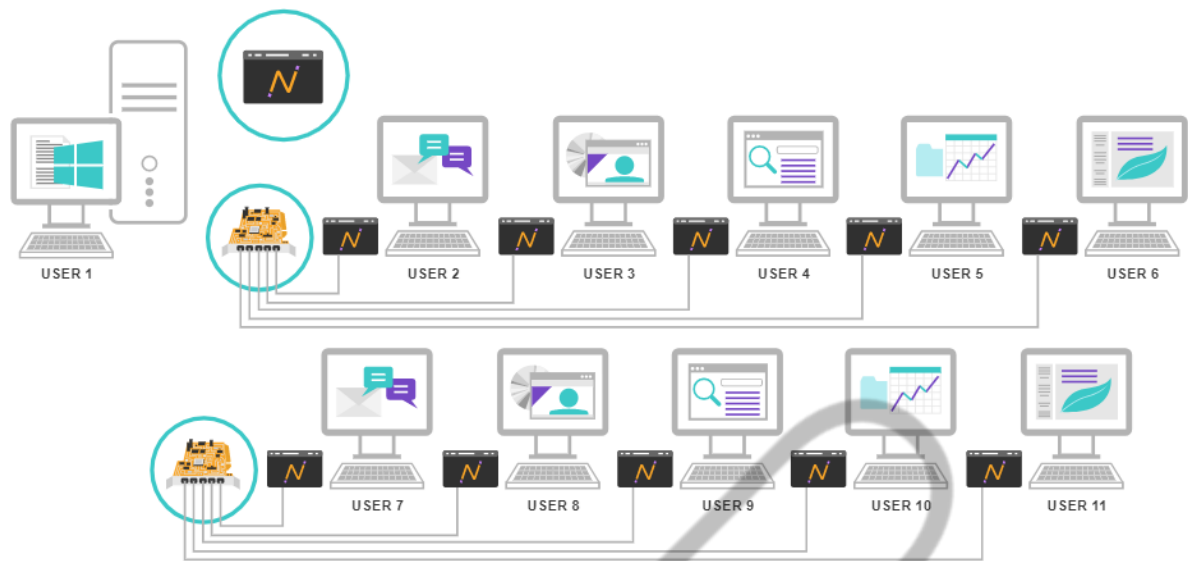
Figura 1.7 – Virtualização de Aplicativos
Fonte: Google Imagens (2018)

Para citar um caso de uso deste tipo de virtualização, lembramos que no início de 2019, **a Google anunciou seu projeto Stadia**, uma plataforma de jogos por streaming. A ideia da empresa é ter um *hardware* de baixo custo a ser “plugado” na TV do assinante (como uma espécie de *Google Chromecast*) apenas para fazer a interface com o *joystick* e rodar a aplicação do serviço; os jogos, que por sua vez requerem um alto poder de processamento, rodarão nos servidores de empresa (VINHA, 2019).
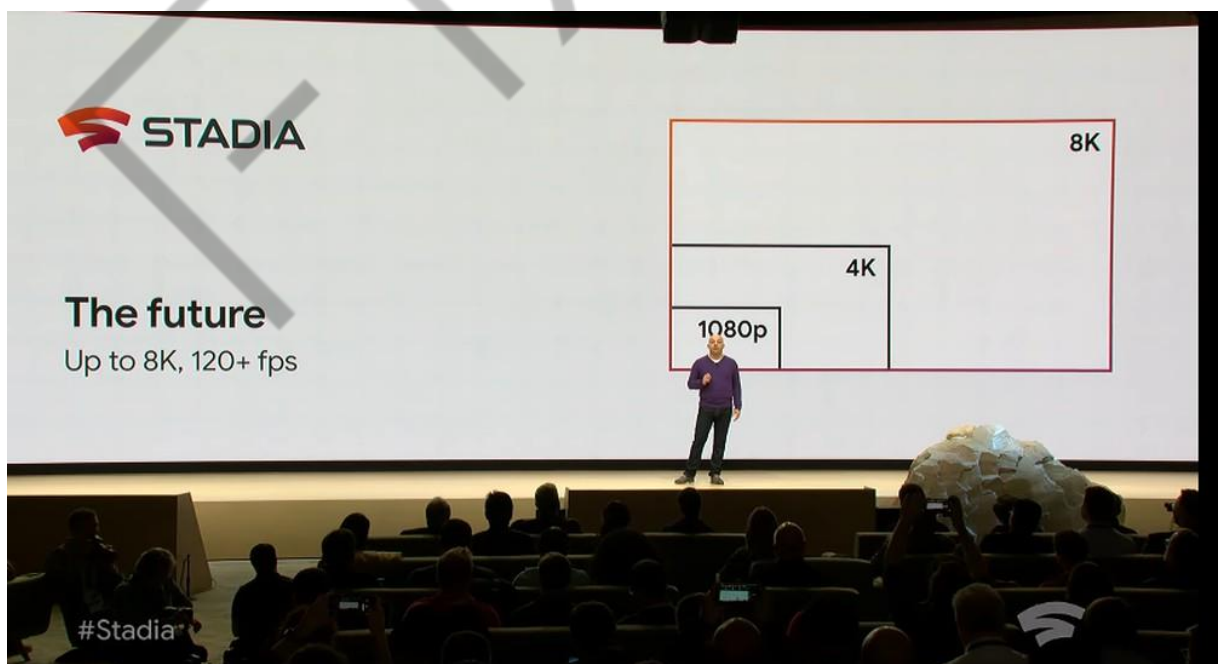
Figura 1.8 – Anúncio do Google Stadia
Fonte: Vinha (2019)

1.4.2 Sistema virtualizado por máquina virtual (H-based)

H-based é o tipo mais comum. Uma máquina virtual (VM, do inglês *Virtual Machine*) requer um Sistema Operacional (SO) completo e exclusivo, além de *kernel* próprio, binários, aplicativos e bibliotecas, o que exige dimensionar espaço grande no servidor e custo de manutenção. Diversas VMs com SOs diferentes podem ser executadas em uma mesma infraestrutura física.
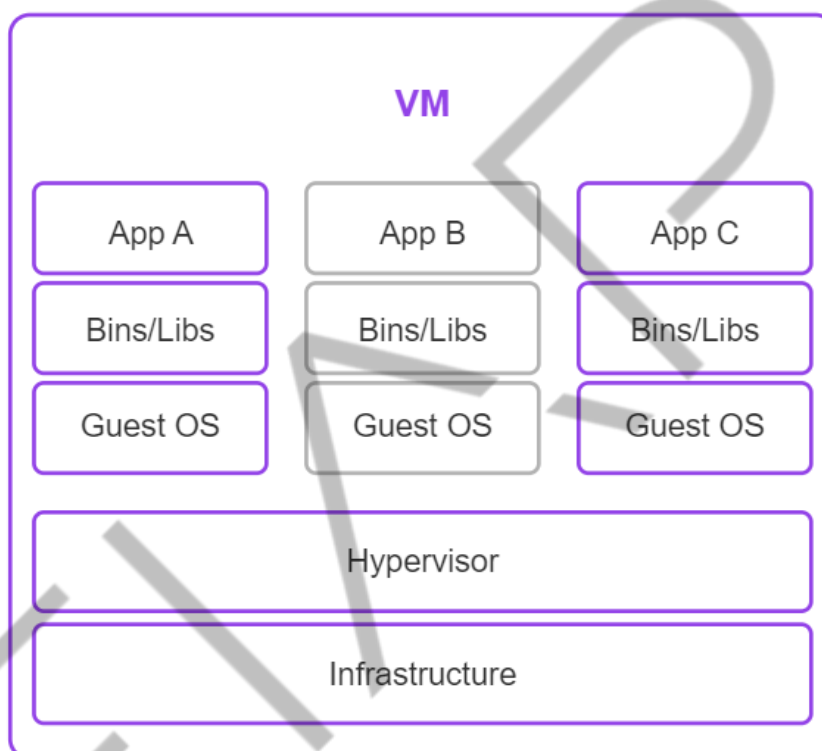
Figura 1.9 – Virtualização por VM
Fonte: DOCKER (s.d.)

Virtualização por VM necessita uma camada intermediária, chamada *hypervisor,* para gerenciar a comunicação de cada VM com o SO hospedeiro (exemplo: VMWare, Hyper-V e o VirtualBox). O *hypervisor* é um elemento fundamental para virtualizar o servidor, pois é responsável por criar e executar VMs, com o intuito de possibilitar que um determinado software seja executado sobre um servidor físico para emular um sistema de hardware.

1.4.3 Sistema virtualizado por container (OS-based)

Implementações de virtualização de *containers* estão fazendo sucesso na *Cloud Computing* devido à facilidade de uso, além de possibilitar melhor gerenciamento de serviços e melhor gestão dos recursos computacionais da nuvem.

“A conteinerização (OS-based) passou a ser amplamente difundida com o surgimento do Docker em 2013” (DOCKER, 2016).
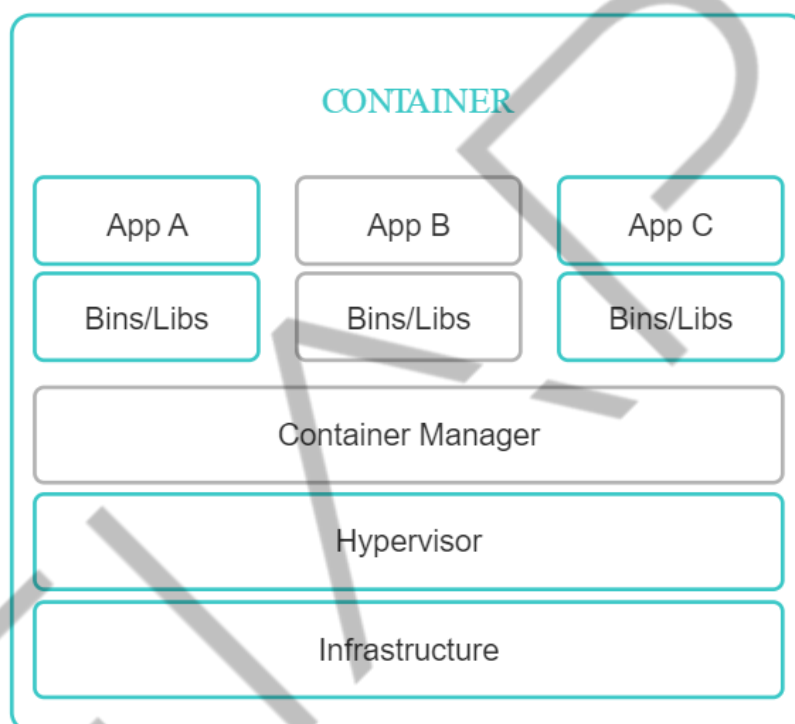
Figura 1.10 – Virtualização por Contêiner
Fonte: DOCKER (s.d.)

Virtualizar por container possibilita, por exemplo, portar sua aplicação diretamente do seu notebook para o servidor de produção ou para uma instância virtual em uma nuvem pública.

A depender da literatura, os containers podem ser chamados de *sandboxes* (ou “caixas de areia”).
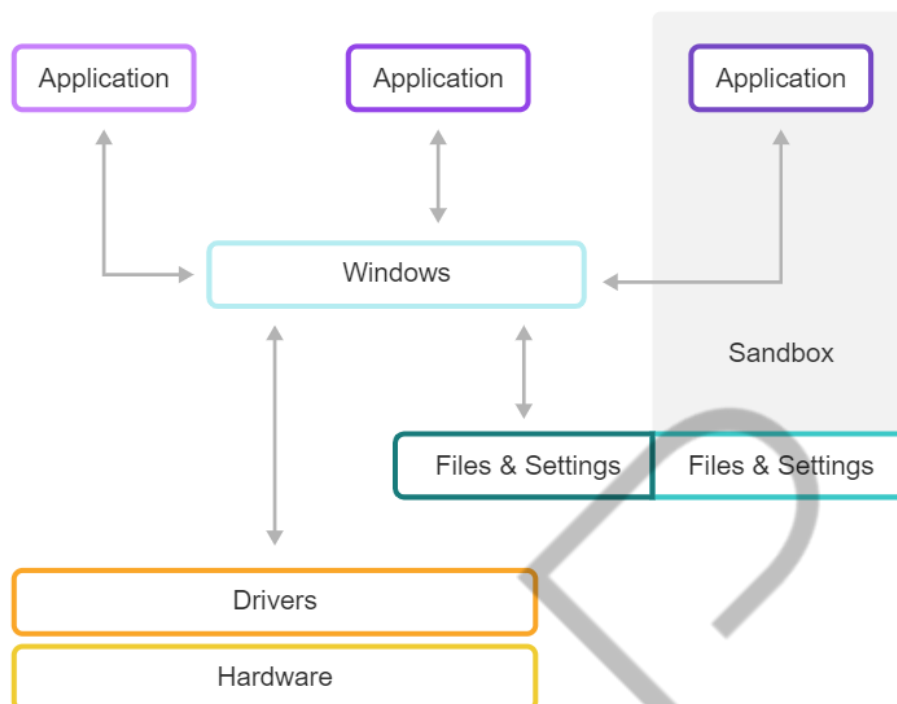
Figura 1.11 – Esquema de uma *sandbox* sobre Windows
Fonte: Notenboom (s.d.)

1.4.4 VM ou Container?

Em suma, a grande diferença entre estes dois tipos de virtualização é que os containers são executados como um processo isolado dentro do *host* e compartilham o *kernel* (SO), enquanto a VM possui um SO completo para cada máquina virtual.

Virtual machine	Container
Desempenho limitado	Desempenho nativo
Virtualização em nível de hardware	Virtualização de SO
Aloca memória necessária	Requer menos espaço de memória
Cada vm é executada em seu próprio SO	Todos os containers compartilham o SO
Tempo de inicialização em minutos	Tempo de inicialização em milissegundos

Quadro 1.1 – Diferenças VM x Container
Fonte: Elaborado pela autora (2019)

- **VM:** é a melhor opção quando se tem uma grande variedade de instâncias de SOs para gerenciar ou quando você precisa executar vários aplicativos em servidores diferentes.
- **Container**: é a melhor opção quando sua prioridade for executar o maior o número de aplicativos em um menor número de servidores.

Comparando a virtualização por container (*OS-based*) com máquinas virtuais (*H-based*), é possível afirmar que a *OS-based* é mais vulnerável nos quesitos isolamento e segurança, mas apresentam melhor desempenho e flexibilidade (ALLES *et al.*, 2018).

Pode acontecer de a sua empresa necessitar da configuração de ambos os tipos de virtualização que podem ser utilizados concomitantemente para fornecer ambientes com funcionalidade máxima. Ambos têm seus benefícios e desvantagens e a decisão final vai depender das necessidades específicas de cada organização.

1.5 Mais sobre o *Hypervisor*

Hypervisor é um grande divisor de águas quando falamos de evolução da computação, pois com ele, vemos uma forma de ultrapassar as possíveis limitações da arquitetura e o alto custo no uso de servidores. Eles podem executar diretamente no *hardware* físico (chamados de *bare-metal*) ou via sistema operacional (*hosted*). Hypervisores são categorizados pelo tipo de modelo de implantação, isto é, pode ser do tipo monolíticos ou microkernelizados.

1.5.1 Hypervisor Monolítico

Nesta figura “Hypervisor Monolitico”, vemos que as VMs são gerenciadas pelo ***hypervisor*** e os *drivers* são hospedados e isso traz alguns tipos de benefícios, por exemplo, o hypervisor, que geralmente não precisará de um controle na partição pai ou no sistema operacional, porque os sistemas operacionais visitantes interagem direto com o hardware utilizando os *drivers* via hypervisor. O desafio é que existem muitos modelos de placas-mãe, placas de vídeo, além de placas de rede e outros dispositivos, dificultando o desenvolvimento específico do hypervisor.
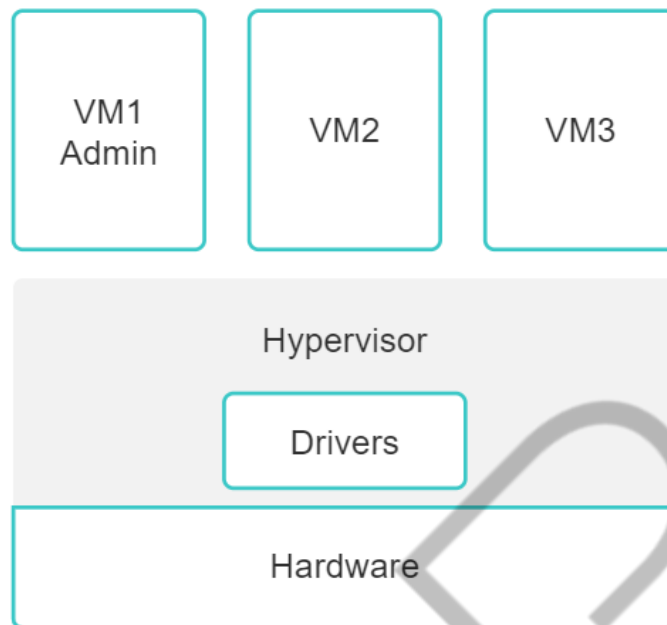
Figura 1.12 – Hypervisor Monolitico
Fonte: Google imagens (2019)

1.5.2 Hypervisor Microkernelizado

No modelo microkernelizado, o hypervisor possui um tipo de S.O. (*Systems Operation*) atuando em forma de *root* (raiz) ou com a partição pai, o hypervisor não necessita de drivers desenvolvidos diretamente para ele. Esta partição fornece um ambiente de execução necessário para os drivers de dispositivo por acessar diretamente o hardware do computador hospedeiro (host). Os drivers dos hardwares físicos são instalados no sistema operacional em execução na partição pai, não sendo necessária a instalação desses drivers nos sistemas operacionais **guests** (convidados). Dessa maneira, quando o sistema operacional convidado precisa acessar o hardware físico, ele simplesmente se comunica com a partição pai.
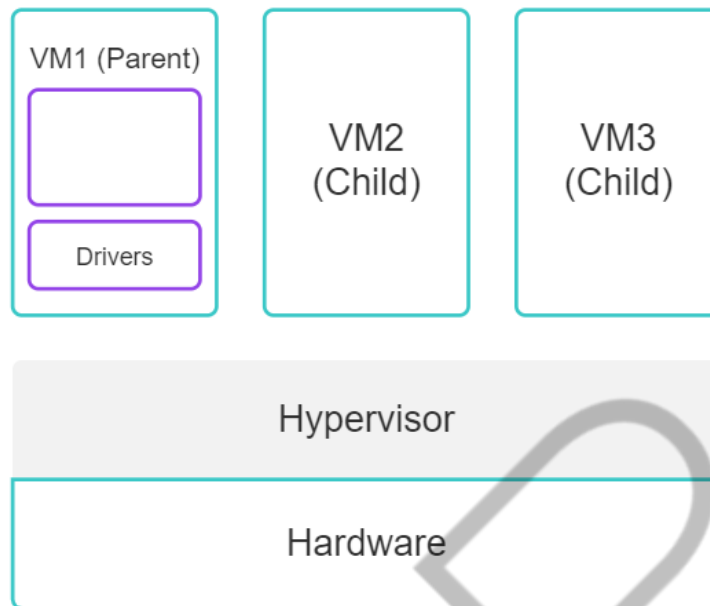
Figura 1.13 – Hypervisor Microkernelizado.
Fonte: Google imagens (2019)

Ainda é possível classificar a virtualização como Virtualização Total (*Full Virtualization*) e Paravirtualização (*Paravirtualization*).

1.5.3 Virtualização total (*Full virtualization*)

O sistema *guest* (convidado) não é capaz de entender se ele está sendo executado em um ambiente virtual. Portanto, se comportará como se estivesse em um ambiente real. Com essa técnica, pode-se instalar o sistema operacional na máquina virtual sem precisar de nenhum tipo de modificação/configuração para que ele funcione corretamente, os recursos necessários são entregues pelo hypervisor na grande maioria das vezes. Essa técnica de virtualização é mais utilizada para virtualizar sistemas operacionais do tipo Windows e MacOS, porque geralmente são de código fechado e não permitem mudanças para reconhecerem que estão em um ambiente virtual. Geralmente, não existe uma ajuda ou facilidade entre convidado (*guest*) e hypervisor.

A grande vantagem é que praticamente não há restrições com o que pode ser virtualizado. Como calcanhar de Aquiles, temos uma redução de performance, justamente porque não existe um trabalho mútuo entre o convidado e o hypervisor). Podemos citar alguns exemplos de softwares que trabalham com essa técnica: KVM, Xen e VMware workstation.
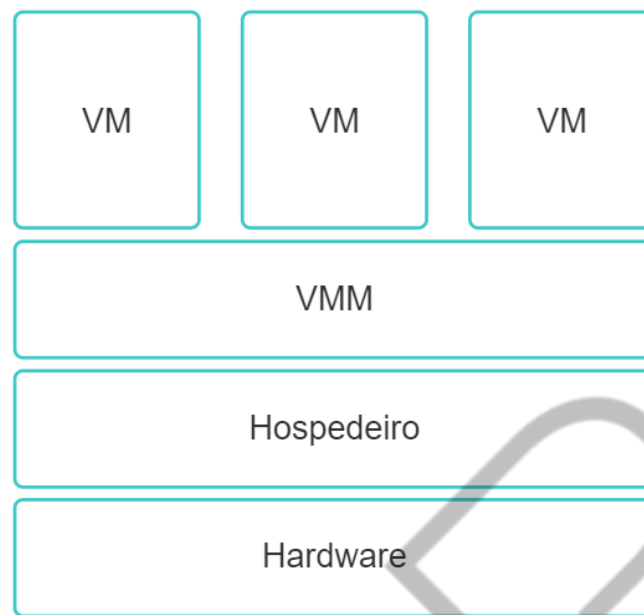
Figura 1.14 – Virtualização total
Fonte: Google imagens (2019)

1.5.4 Paravirtualização (*Paravirtualization*)

Nesta modalidade, o sistema convidado entende que está em um ambiente virtualizado e, desta maneira, as instruções principais ou privilegiadas estão sendo executadas de uma forma diferente, diretamente no *hardware* sem a necessidade de um "tradutor" como o hypervisor. A maioria dos recursos não são todos interpretados como recurso pelo hypervisor, resultando em um ganho de performance especialmente em instruções do tipo I/Os de rede e disco.

Dentre os benefícios dessa técnica, podemos falar sobre a melhor utilização de recursos em oferecer a virtualização, e com relação à performance também é superior a virtualização total.

Como neste caso existe a necessidade de alteração do *kernel*, incluindo e acionando tipos de módulos, não podendo ser realizado em qualquer sistema (somente aqueles que têm código-fonte disponível para alterações). A Microsoft™ encontrou uma maneira de disponibilizar a imagem desse S.O. (Sistema Operacional) que foi especialmente mudada para suportar este tipo de paravirtualização via Xen.
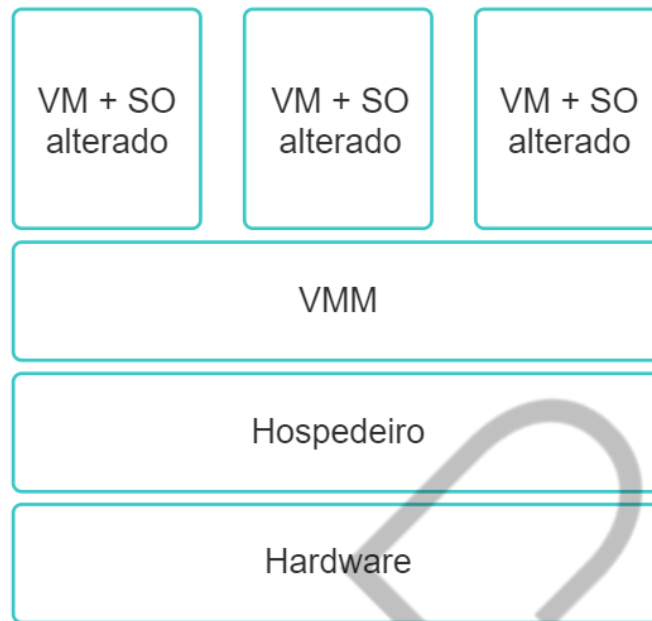
Figura 1.15 – Paravirtualização
Fonte: Google imagens (2019)

1.6 E para o profissional de defesa cibernética, como fica?

Além de compreender melhor os fundamentos da computação em nuvem – o "grande palco" dos ambientes a serem protegidos –, podemos citar uma vantagem adicional muito interessante: a criação de infraestruturas inteiras para testar *softwares* maliciosos (os famosos *malwares*).

Desta forma, conseguimos isolar o software malicioso em um ambiente totalmente selado, da mesma maneira que um epidemiologista estuda um vírus; podemos estudar como ele se propaga e em quais arquivos ele se esconde; quais informações tenta coletar e para quais *hosts* na Internet tenta enviá-las, de forma segura.

Figura 1.16 – Dustin Hoffman no filme “Epidemia”
Fonte: IMDB (s.d)

Como vantagens, podemos destacar:

- O sistema operacional convidado fica isolado do hospedeiro, com isso os *malwares* que rodam na máquina virtual não atingem o host (a menos que exista uma vulnerabilidade 0-day na VM).
- Caso o sistema operacional da VM seja danificado, basta restaurar para o estado original com o recurso de *snapshots*.
- A VM fica contida em poucos arquivos, tornando mais fácil o seu *backup*, duplicação, entre outros.

A seguir, montamos um ambiente para execução deste tipo de teste. Primeiro, devemos instalar uma plataforma de virtualização. Uma boa sugestão é o **Oracle VirtualBox**, que é gratuita e de fácil utilização.

Baixe-a em: <https://www.virtualbox.org/wiki/Downloads>.
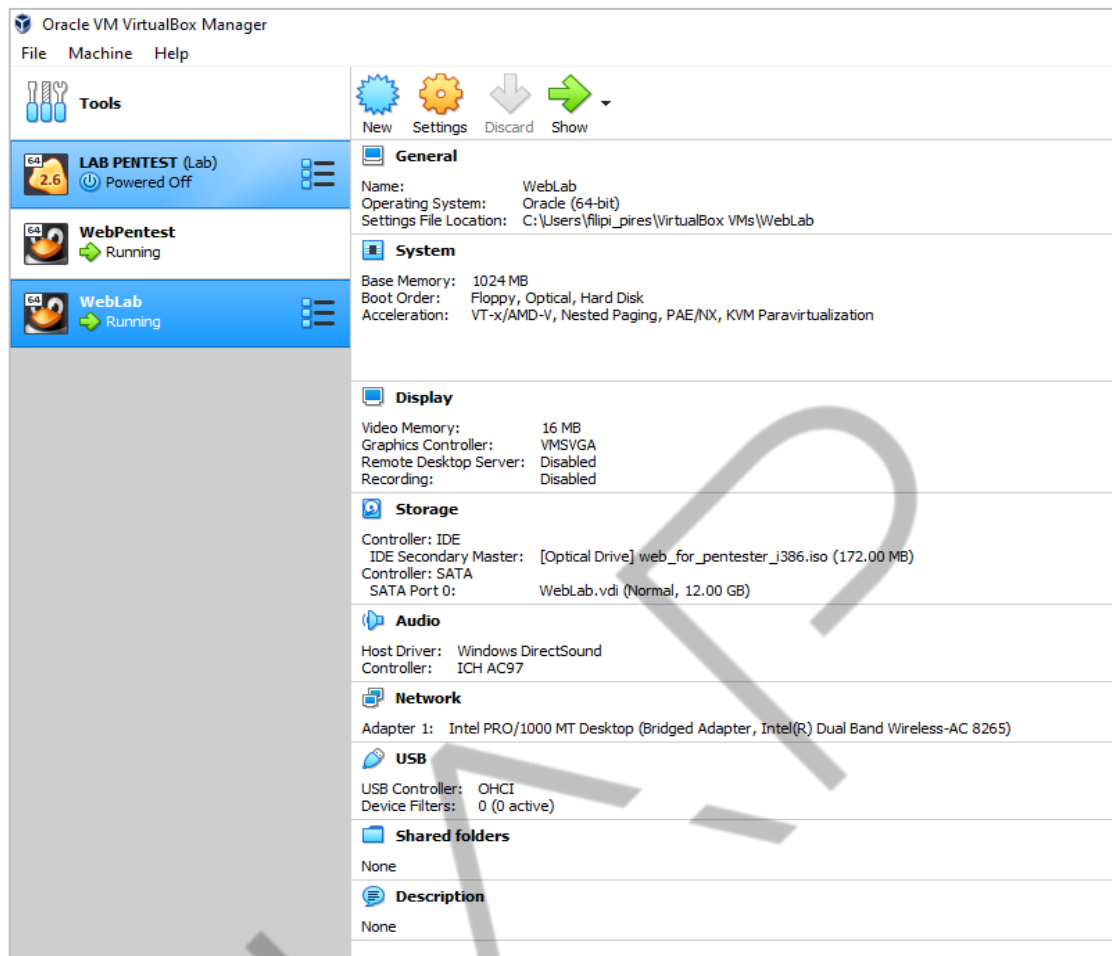
Figura 1.17 – Oracle Virtual Box
Fonte: Elaborado pelo autor Filipe (2019)

Outra alternativa é o **VMWare Workstation**, que possui uma interessante série de recursos. Sua versão PRO pode ser baixada na versão *trial*, no entanto, depois do período de testes, é necessário adquiria sua licença.

Pode ser baixado em: <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>.
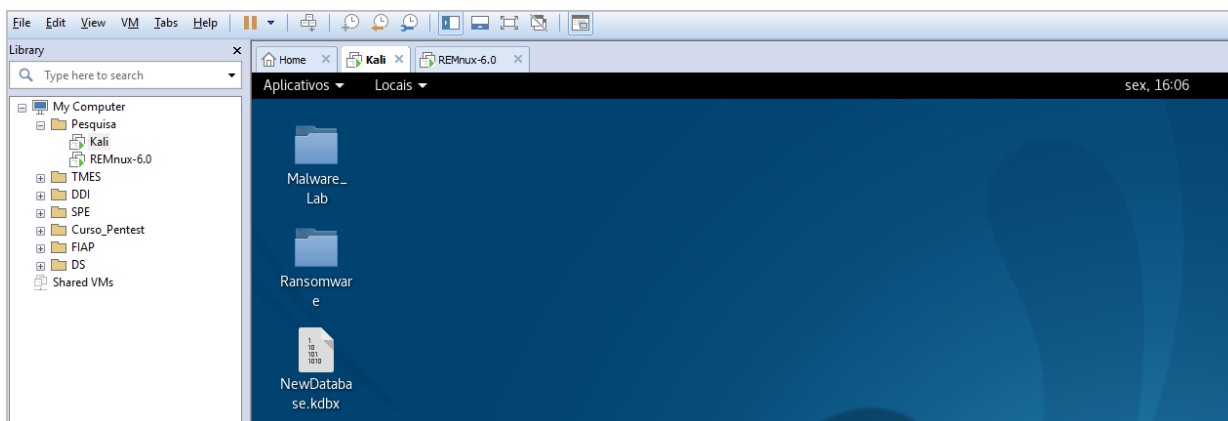
Figura 1.18 – VMWare Workstations 12 PRO
Fonte: Elaborado pelo autor (2019)

Quando lidamos com virtualização, temos tomar alguns cuidados mesmo em máquina virtuais. Em primeiro lugar, devemos manter o *software* de virtualização atualizado, pois além das atualizações fornecerem recursos adicionais para a VM, elas são utilizadas para corrigir falhas de segurança da plataforma. A atualização mantém as VMs livres das vulnerabilidades conhecidas, assim o *malware* só conseguirá "escapar" da máquina virtual através de uma vulnerabilidade 0-day.

Outra questão crítica à qual devemos estar atentos diz respeito à conexão de rede da máquina virtual. Caso ela esteja mal configurada, o *malware* poderá infectar o computador hospedeiro e até outros *hosts* da rede caso esteja em uma LAN. Em um cenário menos crítico, pode se tornar um *bot* para atacar máquinas de terceiros, enviar *spams*, hospedar conteúdo ilegal, entre outras possibilidades que não parecem nada interessantes.

1.6.1 Configurando a rede

Lidar com a rede em máquinas virtuais requer o entendimento das configurações de rede que são oferecidas nas plataformas de virtualização mais comuns.

A descrição dos três modos é feita a seguir:

- ***Host-only***: Cria uma LAN privada compartilhada no hospedeiro e suas VMs. VMs não podem se comunicar com computadores externos.
- **NAT/Shared:** VMs podem acessar outros computadores da LAN ou Internet, mas as conexões aparecerão vindas do IP do hospedeiro. Os outros

computadores não podem iniciar conexões com a VM, a menos que seja configurado um redirecionamento de portas (*port-forwarding*) no computador hospedeiro.

- **Bridged:** VMs compartilham o adaptador Ethernet físico do hospedeiro, mas possuem seus próprios endereços IPs e MACs. As VMs aparecem na mesma sub-rede (*subnet*) do hospedeiro. Essa é a única configuração que permite quer outros computadores iniciem conexões de entrada na VM. É também o único modo que permite que outras máquinas externas, por exemplo, um roteador ou *firewall*, distingam o tráfego gerado pelo hospedeiro do tráfego das VMs.

Sintetizamos questões importantes no quadro "Modos de rede e suas possibilidades".

Acesso	*Host-only*	*NAT*	*Bridged*
VMs podem acessar outras VMs	Sim	Sim	Sim
VMs podem acessar o hospedeiro	Sim	Sim	Sim
VMs podem acessar outros computadores	Não	Sim	Sim
O hospedeiro pode acessar VMs	Sim	Sim	Sim
Outros computadores podem acessar VMs	Não	Não	Sim

Quadro 1.2 – Modos de rede e suas possibilidades
Fonte: Adaptado de LIMA (s.d.)

Escolhendo um desses modos, as máquinas virtuais permitem a simulação do cabo de rede conectado ou desconectado. É recomendado que enquanto não estiver sendo utilizada nenhuma função de rede, o cabo de rede permaneça no modo **desconectado**, o qual somente deverá ser ativado quando necessário. No VMWare, isso é feito no menu Virtual Machine – Removable Devices; no Network Adapter no VirtualBox é através do menu Dispositivos – Adaptadores de rede.

Vamos a algumas considerações que são importantes no momento da construção da rede de um ambiente virtual. Em uma análise, a comunicação do *malware* com a rede é algo essencial de ser observado, porém não se deve permitir que um *malware* desconhecido faça isso de forma indiscriminada conectando-o diretamente à Internet. O recomendado é que se configure o adaptador de rede em modo ***host-only*** e que sejam realizadas as análises iniciais do artefato para, só posteriormente, ampliar a conectividade e deixá-lo, bem, "mais saidinho".

O momento certo para nos aprofundar nestes estudos ainda vai chegar. Este breve relato envolvendo a análise de *malwares* em máquinas virtuais foi realizado apenas para contextualizar o quão importante esta ferramenta será para nossos estudos. Espero que esta espiadinha (ou como os americanos diriam, *sneak peak*) tenha deixado você com água na boca.

1.7 Dicas finais

Vamos a algumas dicas finais que podem aprimorar nossa usabilidade com as máquinas virtuais.

1.7.1 Recursos Adicionais da VM

Tanto o VMWare quando o VirtualBox oferecem a opção de instalar recursos adicionais na máquina virtual, que facilitam a forma com *que* o sistema hospedeiro interage com a máquina virtual, especialmente no compartilhamento de arquivos entre os dois. Eles permitem, entre outras possibilidades, instalar um driver para a placa de vídeo virtual que a plataforma emula, permitindo visualizar um *desktop* em “tela cheia”.

Então é recomendada a instalação desses recursos, no VMWare chama-se “**VMWare Tools**” e no VirtualBox “**Adicionais para convidados**”.

1.7.2 Pastas Compartilhadas

Um recurso que pode ser utilizado para a troca de arquivos entre o computador hospedeiro e a máquina virtual é o de pastas compartilhadas. Permite mapear uma pasta do hospedeiro dentro da máquina virtual, assim tudo o que for colocado nessa pasta estará disponível para os dois sistemas.

Como medida de segurança caso utilize esse recurso, marque a pasta compartilhada como sendo **somente-leitura**, assim a máquina virtual não conseguirá gravar nada nela e quando necessário habilite a gravação. Todo cuidado é pouco!

1.7.3 Snapshots

Tirar *snapshots* é um conceito único das máquinas virtuais e é absolutamente obrigatório para o profissional de segurança cibernética. É este recurso que nos permitirá salvar o estado atual da VM e restaurá-lo posteriormente sempre que for necessário.

Conforme exemplificado neste capítulo, será muito útil na análise de *malware* pois poderemos fazer com que o software malicioso repita seus procedimentos em uma situação controlada, restaurando o sistema quantas vezes quisermos e quando quisermos.

CONCLUSÃO

Em suma, a virtualização é o coração da computação em nuvem (*Cloud Computing*) e responsável por esta grande revolução. Afinal, é graças a essa e outras tecnologias que *startups* no mundo inteiro ganham escalabilidade na velocidade que tanto precisam.

Considerando a importância da infraestrutura empresarial baseada em *cloud,* é nesse tipo de estrutura que a maioria dos ambientes a se proteger estarão no futuro, tornando este aprendizado obrigatório para um profissional de segurança da informação, cujas máquinas virtuais são uma de suas mais importantes ferramentas.

REFERÊNCIAS

ALECRIM, E. **O que é virtualização e para que serve?** 2012. Disponível em: <https://www.infowester.com/virtualizacao.php>. Acesso em: 8 fev. 2021.

ALLES, G. R.; CARISSIMI, A.; SCHNORR, L. M. Assessing the computation and communication overhead of linux containers for hpc applications. **Anais do Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD**), 2018. Disponível em: <https://www.researchgate.net/publication/334168150_Assessing_the_Computation_and_Communication_Overhead_of_Linux_Containers_for_HPC_Applications/link/5d3617d24585153e59196410/download>. Acesso em: 8 fev. 2021.

BRAGA, A. S.; SILVA, G. M.; BARROS, M. C. **Cloud Computing**. Disponível em: <http://www.ic.unicamp.br/~ducatte/mo401/1s2012/T2/G08-079713-079740-820650-t2.pdf>. Acesso em: 8 fev. 2021.

DOCKER. **Docker Get Started**. [s.d.]. Disponível em: < https://docs.docker.com/get-started/>. Acesso em: 9 out. 2019.

DOCKER. **Modern app architecture for the enterprise.** 2016. Disponível em: <https://www.docker.com/sites/default/files/caaSwhitepaper_V6_0.pdf>. Acesso em: 8 fev. 2021.

IMDB. **Epidemia.** (1995). Disponível em: <https://www.imdb.com/title/tt0114069/mediaviewer/rm1151892992>. Acesso em: 12 fev. 2021.

INFOWESTER. **Uso Ócio**. [s.d.]. Disponível em: <https://www.infowester.com/img_art/uso_ocio.jpg>. Acesso em: 9 out. 2019.

LIMA, R. P. de. **Engenharia Reversa e Análise de Malwares.** [s.d.]. Disponível em: <https://pt.scribd.com/document/406971131/Slides-Aula-04-pdf>. Acesso em: 8 fev. 2021.

NOTENBOOM, Leo A. **What's the Difference Between a Sandbox and a Virtual Machine?** [s.d.]. Disponível em: <https://askleo.com/whats-the-difference-between-a-sandbox-and-a-virtual-machine/>. Acesso em: 9 out. 2019.

NUBLING, G. **Cloud Computing aplicada ao Cenário Corporativo.** Trabalho de Conclusão de Curso (Tecnologia em Processamento de Dados) – Faculdade de Tecnologia de São Paulo, 2011. Disponível em: <http://www.fatecsp.br/dti/tcc/tcc0038.pdf>. Acesso em: 8 fev. 2021.

RADOSLAV, C. **Cloud Computing Statistics 2019.** 2021. Disponível em: <https://techjury.net/stats-about/cloud-computing/>. Acesso em: 8 fev. 2021.

RUPARELLA, N. **Cloud Computing**. Londres: Th MIT Press, 2016.

VELTE, A. T.; VELTE, J. T.; ELSENPETER, R. **Cloud Computing** – A Practical Approach. New York: The McGraw-Hill Companies, Inc., 2010.

VINHA, F. **Google anuncia Stadia, plataforma de jogos por streaming.** 2019. Disponível em: <https://www.techtudo.com.br/noticias/2019/03/google-anuncia-stadia-servico-de-games-por-streaming.ghtml>. Acesso em: 8 fev. 2021.

VIRTUALIZAÇÃO – Definição. **Escotilha Livre**, 2015. Disponível em: <https://escotilhalivre.wordpress.com/2015/06/10/virtualizacao-definicao>. Acesso em: 8 fev. 2021.